Software Engineering Institute

Software Product Lines: Report of the 2010 U.S. Army Software Product Line Workshop

John K. Bergey
Gary Chastek
Sholom Cohen
Patrick Donohoe
Lawrence G. Jones
Linda Northrop

June 2010

TECHNICAL REPORT
CMU/SEI-2010-TR-014
ESC-TR-2010-014

Research, Technology, and System Solutions Program

http://www.sei.cmu.edu

Carnegie Mellon

This report was prepared for the

SEI Administrative Agent
ESC/XPK
5 Eglin Street
Hanscom AFB, MA 01731-2100

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

For information about SEI publications, please visit the library on the SEI website (http://www.sei.cmu.edu/library).

Table of Contents

Acknowledgments **v**

Abstract **vii**

1 Introduction **1**
1.1 Product Line Practice 1
1.2 About this Workshop 2
1.3 About this Report 3

2 Software Product Line Experiences: A Digest of Participant Presentations **5**
2.1 Introduction – Linda Northrop, SEI 5
2.2 A Conceptual View of a Software Product Line Acquisition – John Bergey, SEI 5
2.2.1 Basic Product Line Acquisition Strategies 5
2.2.2 Contractual Tasks for a Software Product Line Acquisition 5
2.2.3 An Example of a Work Breakdown Structure for a Product Line Acquisition 6
2.2.4 An Enterprise View of a Software Product Line Acquisition 7
2.2.5 A Customer View of a Software Product Line Acquisition Enterprise 8
2.3 Joint Fires Product Line – Glen Loupe, PEO STRI 9
2.3.1 Product Composition and Documentation 10
2.3.2 Concept of Operations 11
2.3.3 Adoption Challenges and Lessons Learned 12
2.4 Extending the Live Training Transformation Product Line – Bob Becker, Jacobs Engineering 13
2.5 Common Driver Trainer Product Line – Dean Runzel, PEO STRI 14
2.6 Common Link Integration Processing Product Line – Gary Newman, Northrop Grumman 16

3 A Summary of the Facilitated Discussions **19**
3.1 Discussion Topic: What Measures Do We Need to Demonstrate Product Line Value? 19
3.2 Discussion Topic: What's in It for the Supplier? What Is Needed to Incentivize Suppliers? 20
3.3 Discussion Topic: What Technical Evaluation Criteria Should the Government Use? 21

4 Summary **23**

References/Bibliography **25**

List of Figures

Figure 1:	Three Major Contractual Tasks for a Software Product Line Acquisition	6
Figure 2:	Sample Product Line Developer Work Breakdown Structure (WBS)	7
Figure 3:	Sample Enterprise View of a Product Line Acquisition	8
Figure 4:	Simplest Case of Customer Interaction with Product Line Developer	9

Acknowledgments

The workshop organizers would like to acknowledge the work of Barbara Pemberton, Program Executive Office, Simulation Training and Instrumentation (PEO STRI), in hosting this workshop, and Carolyn Kernan of the SEI for being the workshop coordinator. Thanks also to Bob Schwenk and the Army Strategic Software Improvement Program (ASSIP) for their support of these workshops.

Abstract

The Carnegie Mellon Software Engineering Institute held the U.S. Army Software Product Line Workshop on February 11, 2010. The workshop was a hands-on meeting to share Army and Department of Defense product line practices, experiences, and issues and to discuss specific product line practices and operational accomplishments. Participants reported encouraging progress on Army software product lines. This report synthesizes the workshop presentations and discussions.

1 Introduction

1.1 Product Line Practice

A *software product line* is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [Clements 2002]. An increasing number of organizations are building their products as product lines in order to achieve large-scale productivity gains, improve time to field or market, maintain a market presence, compensate for an inability to hire, leverage existing resources, and achieve mass customization.

In January 1997, the Carnegie Mellon® Software Engineering Institute (SEI) launched the Product Line Practice Initiative to help facilitate and accelerate the transition to sound software engineering practices using a product line approach. The goal of this initiative is to provide organizations with an integrated business and technical approach to systematic reuse, so they can produce and maintain similar systems of predictable quality more efficiently and at a lower cost.

A key strategy for achieving this goal has been the creation of a framework for product line practice. The SEI *Framework for Software Product Line Practice*℠ (henceforth referred to as "the framework") describes the foundational product line concepts and identifies the essential activities and practices that an organization must master before it can expect to successfully field a product line of software or software-intensive systems. The framework is a living document that is evolving as experience with product line practice grows. Version 4.0 is described in the book *Software Product Lines: Practices and Patterns* [Clements 2002], and the latest version is available on the SEI website [Northrop 2010].

The framework's contents are based on information-gathering workshops,[1] extensive work with collaboration partners, surveys and investigations, and continued research. The SEI has also incorporated practices reported at its international Software Product Line Conferences and collected information from the community [Donohoe 2000, Chastek 2002, Nord 2004, Obbink 2005, O'Brien 2006, IEEE 2007, Geppert 2008, McGregor 2009b].

In March 1998, the SEI hosted its first Department of Defense (DoD) product line practice workshop [Bergey 1998]. Topics discussed and documented included DoD barriers and mitigation strategies, and similarities and differences between DoD and commercial practice. Subsequent workshops were held in successive years [Bergey 1999, 2000a, 2001, 2003, 2004, 2005a, 2005b, 2009].

At each of these, the SEI was encouraged to continue holding DoD workshops and to continue sharing best commercial and DoD product line practices through these forums. In 2006, the work-

® Carnegie Mellon is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

SM Framework for Software Product Line Practice is a service mark of Carnegie Mellon University.

1 The results of some of these workshops are documented in SEI reports [Bass 1997, 1998, 1999, 2000; Clements 2001].

shop was held as a “birds of a feather” session in conjunction with the International Software Product Line Conference (SPLC 2006) in Baltimore, Maryland. In 2007, sponsorship of the workshops switched to the Army Strategic Software Improvement Program (ASSIP) under the auspices of the Assistant Secretary of the Army for Acquisition, Logistics and Technology [ASA (ALT)].

1.2 About this Workshop

The workshop goals were to

- share Army and DoD product line practices, experiences, and issues, from both development and acquisition viewpoints
- examine barriers and enablers to much broader adoption of software product line practices within the Army and the DoD
- determine the steps needed to make software product line practices more beneficial and relevant to Army and DoD acquisition programs
- discuss ways in which the Army’s Strategic Software Improvement Program (ASSIP) can be of assistance

Almost all participants in this workshop were from the DoD acquisition and contractor community. They were invited based on our knowledge of their experience with and commitment to software product lines as either DoD system acquirers or DoD system contractors. Together, the group discussed the issues that form the backbone of this report.

The format of this workshop followed that of the previous successful workshops. Invited presentations were followed by a facilitated discussion of ideas stimulated by the presentations. The group agreed that this format worked well.

The workshop participants included

- Bob Becker, Jacobs Engineering
- John Bergey, SEI
- Eric Byrd, U.S. Army
- Gary Chastek, SEI
- Myra Cohen, University of Nebraska
- Sholom Cohen, SEI
- Patrick Donohoe, SEI
- Jerry P. Ervin, General Dynamics C4 Systems
- Terry Gatewood, U.S. Army
- David Grow, USA PM ITTS
- David A. Hill, L-3 Services, Inc., C2S2
- Jack Hine, Jacobs Technology
- Larry Jones, SEI

- Todd Kohler, U.S. Army
- Keith W. Lane, Northrop Grumman
- Glen Loupe, PEO STRI - PM STS
- Christal Martir, Lockheed Martin (CTIA)
- Roger McNicholas, General Dynamics C4 Systems
- Ray Menell, CECOM LCMC SEC
- Bryce L. Meyer, SEI
- Gary Newman, Northrop Grumman
- Khuc Nguyen, PEO-STRI
- Linda Northrop, SEI
- Don O'Connell, Boeing
- Roger Olson, Nova Technologies
- Barbara J. Pemberton, PEO STRI
- Todd Peterson, GDC4S
- Ellen Reinig, Programmatics Engineering Group
- Hal Roby, USMC Systems Command, PM TRASYS, Encomium Research
- Dean Runzel, PEO STRI
- David Schuerer, SRI International
- Scott Szurgot, PM TRASYS
- Damla Turgut, University of Central Florida
- Cisca Vuong, PEO STRI
- David Wade, U.S. Army

1.3 About this Report

This document summarizes the presentations and discussions from the workshop. This report is written primarily for those in the DoD who are already familiar with product line concepts, especially those working on or initiating product line practices in their own organizations. Acquisition managers and technical software managers should also benefit from this report. Those who desire further background information are referred to the following resources:

- *Software Product Line Essentials* [Northrop 2008]
- *Basic Concepts of Product Line Practice for the DoD* [Bergey 2000b]
- *Product Line Acquisition in a DoD Organization—Guidance for Decision Makers* [Bergey 2006]
- *A Framework for Software Product Line Practice, Version 5.0* [Northrop 2010]
- *Software Product Lines: Practices and Patterns* [Clements 2002]

The next section of this report contains a digest of the presentations. A summary of the facilitated discussion follows. The report concludes with a brief summary.

2 Software Product Line Experiences: A Digest of Participant Presentations

2.1 Introduction – Linda Northrop, SEI

Linda Northrop, Director of the Research, Technology, and System Solutions Program at the SEI, began by explaining the workshop goals and agenda. She then gave an overview of software product line practice. As noted earlier, readers who would like an explanation of the basics of software product lines should see the references in Section 1.3.

2.2 A Conceptual View of a Software Product Line Acquisition – John Bergey, SEI

John Bergey's presentation provided a conceptual example of the allocation of responsibilities and stakeholder collaborations necessary to execute a successful product line acquisition strategy. A software product line acquisition presents some unique challenges to DoD programs and the contractors responsible for their development. It involves adopting new practices, specifying an appropriate division of responsibilities, and contracting with suppliers to manage, develop, operate, and sustain a product line. Best practice is to specify the product line aspects up front—as opposed to opportunistically attempting to initiate a product line approach under an existing contract—so that an appropriate set of requirements and statement of work (SOW) tasks can be included in the request for proposal (RFP) and the contract.

2.2.1 Basic Product Line Acquisition Strategies

Developing a suitable acquisition strategy is a key consideration in adopting a product line approach in the DoD. There are three basic strategies for acquiring software products via a product line:

1. A program management office commissions a contractor to develop products using the contractor's proprietary software product line.
2. A program management office commissions a government organization to develop a software product line.
3. A program management office commissions a contractor to develop a government-owned software product line.

The difficulty in executing these strategies varies significantly, since they require different levels of management sophistication and technical skills on the part of the acquisition organization. Of the three approaches listed, the third is the most challenging.

2.2.2 Contractual Tasks for a Software Product Line Acquisition

At the highest level of abstraction, a software product line acquisition consists of three contractual tasks (Figure 1) to be performed by the developer:

1. development of a product line production capability
2. development of a family of software products using that production capability
3. management and operation of the product line

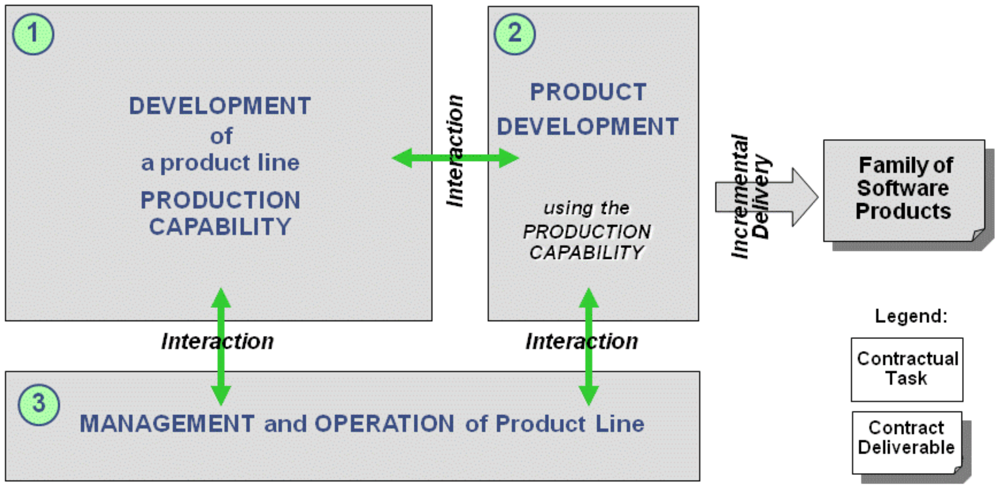

Figure 1: Three Major Contractual Tasks for a Software Product Line Acquisition

Developing a product line production capability includes providing the product line core assets and a production plan to enable products to be built in a prescribed way. A software development plan, which is a traditional contractual document required by the DoD, can be used to describe (and govern) the actual development of such a product line production capability.

Developing a product means using the production capability and its core assets to develop a specific product as a member of the product family, according to a documented production plan. The production plan identifies the techniques to be used, the schedule for using them, and the materials needed to build the product.

Managing the product line means following through on several plans, including, for example, a product line adoption plan and a core asset funding plan. Operating the product line includes implementing a product line concept of operations that describes how organizational roles and responsibilities (e.g., product line manager, core asset developer, and product developer) interact to achieve the goals established for the product line.

2.2.3 An Example of a Work Breakdown Structure for a Product Line Acquisition

Figure 2 provides a sample partial work breakdown structure (WBS) that corresponds to the three contractual product line SOW tasks in Figure 1. Two additional tasks (at the third tier level) account for sustaining the production capability over the life cycle and sustaining fielded products that are in operational use. While there is no one-size-fits-all, the WBS example serves as a useful starting point that an acquisition organization can appropriately expand and tailor to meet its specific needs.

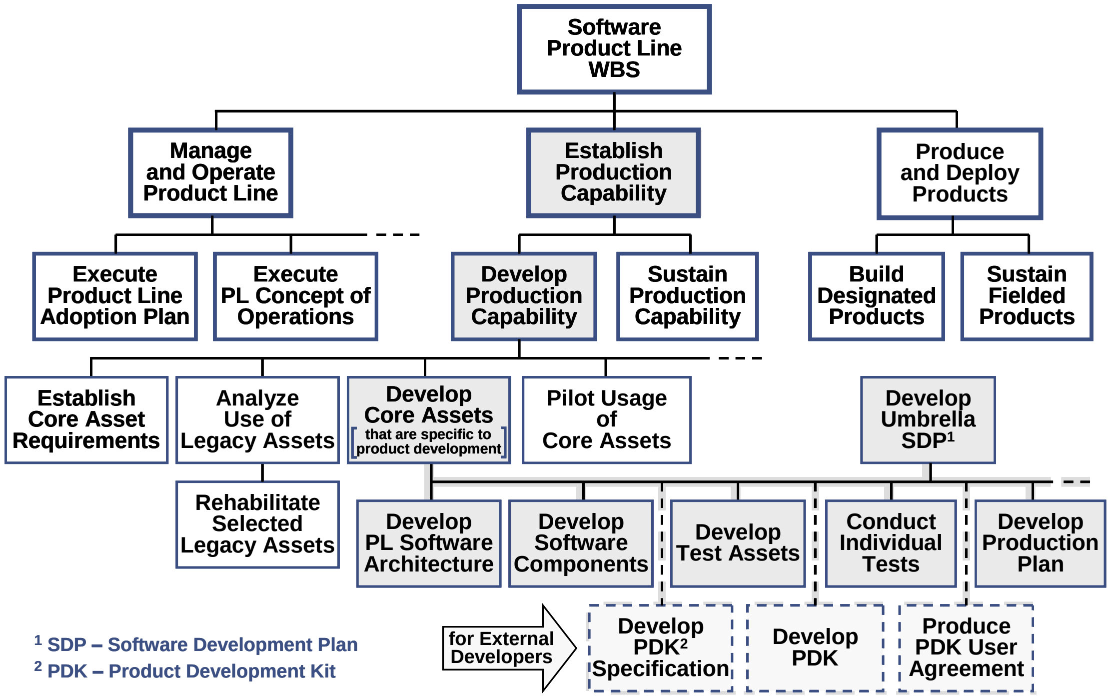

Figure 2: Sample Product Line Developer Work Breakdown Structure (WBS)

To gain a proper perspective of the activities and tasks that a product line initiative involves, it is useful to create a context diagram in the form of an enterprise[2] view, which is described in the next sections.

2.2.4 An Enterprise View of a Software Product Line Acquisition

The life cycle of a product line encompasses a cast of characters and relationships larger than is typical for a single system. This idea extends to product line acquisition. A major benefit of a product line acquisition enterprise view is that it provides additional insight into what is involved in an acquisition context and motivates the acquirer and developer to question—at a deeper level—how they plan to collaboratively manage and operate the product line. It is also a valuable tool for providing affected stakeholders with a common understanding of how the product line acquisition will be implemented.

Figure 3 depicts a notional view of a software product line acquisition enterprise that corresponds to the third acquisition strategy described in Section 2.2.1. This sample enterprise view captures the essence of the major organizational elements and activities in an acquisition context and helps ensure that all stakeholders have a common understanding of the ramifications of the adopted approach.

The two primary organizational elements in this view are the parent government organization, which is responsible for acquiring the product line, and the prime contractor's organization, which is responsible for implementing and sustaining the product line.

---

2 The original workshop presentation had an "ecosystem" view and an accompanying set of animated slides. This report uses the enterprise view and an accompanying set of figures better suited to the report format. For more information on the ecosystem concept for product lines, see the paper by McGregor [McGregor 2009a].

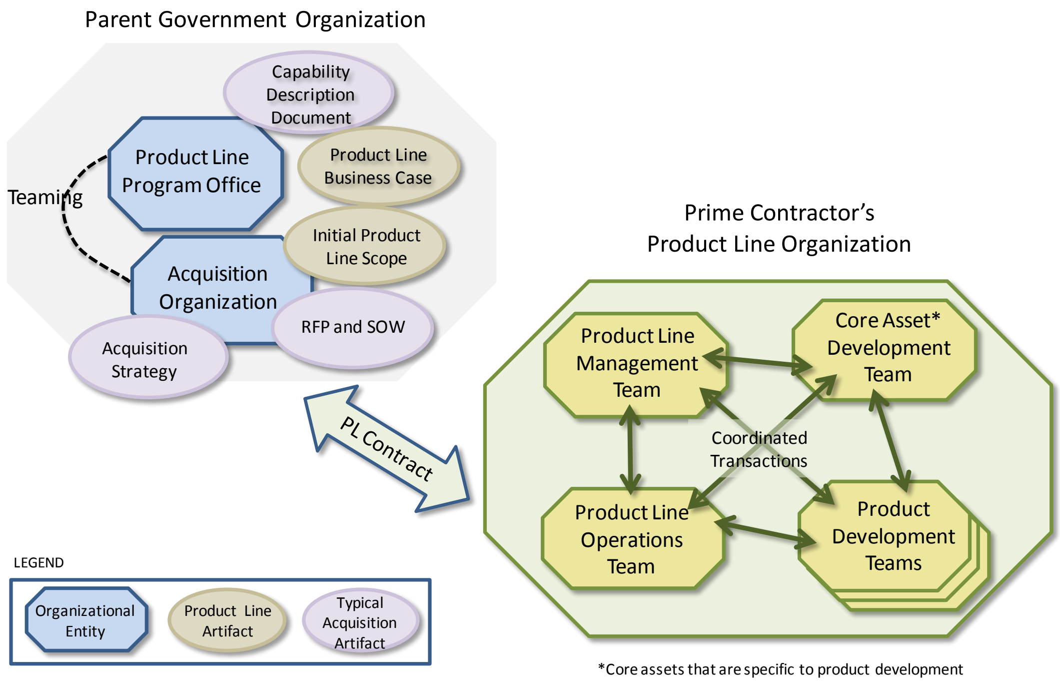

Figure 3: Sample Enterprise View of a Product Line Acquisition

The breakdown of the prime contractor's product line organization into a management team**,** core asset team, product development team, and operations team is just one example of how a developer organization might implement a product line approach. In this configuration, the management, core asset, and operations teams are the organizational elements that are responsible for establishing the production capability that the product development teams will use. Product development teams would be responsible for the key deliverables associated with the particular product they are developing, such as a product requirements specification, product-unique software components, and product test artifacts and plans.

2.2.5 A Customer View of a Software Product Line Acquisition Enterprise

There is also a customer view of the acquisition enterprise, as shown in Figure 4. The object of this view is to show how a customer would interact with the product line. While there are several potential customer views, this one depicts the simplest case, where the program office is also the customer. The program office is the customer if the target system that incorporates the software product is under the jurisdiction of the program office. If the customer is not the program office, which would often be the case, the interactions naturally become more complex.

While the program office is ultimately responsible for both the product and the system to which it belongs, a system prime contractor (under contract to the program office) is the agent that is actually responsible for developing and sustaining the target system. This situation corresponds to what Figure 4 identifies as the parent organization and the expanded customer environment.

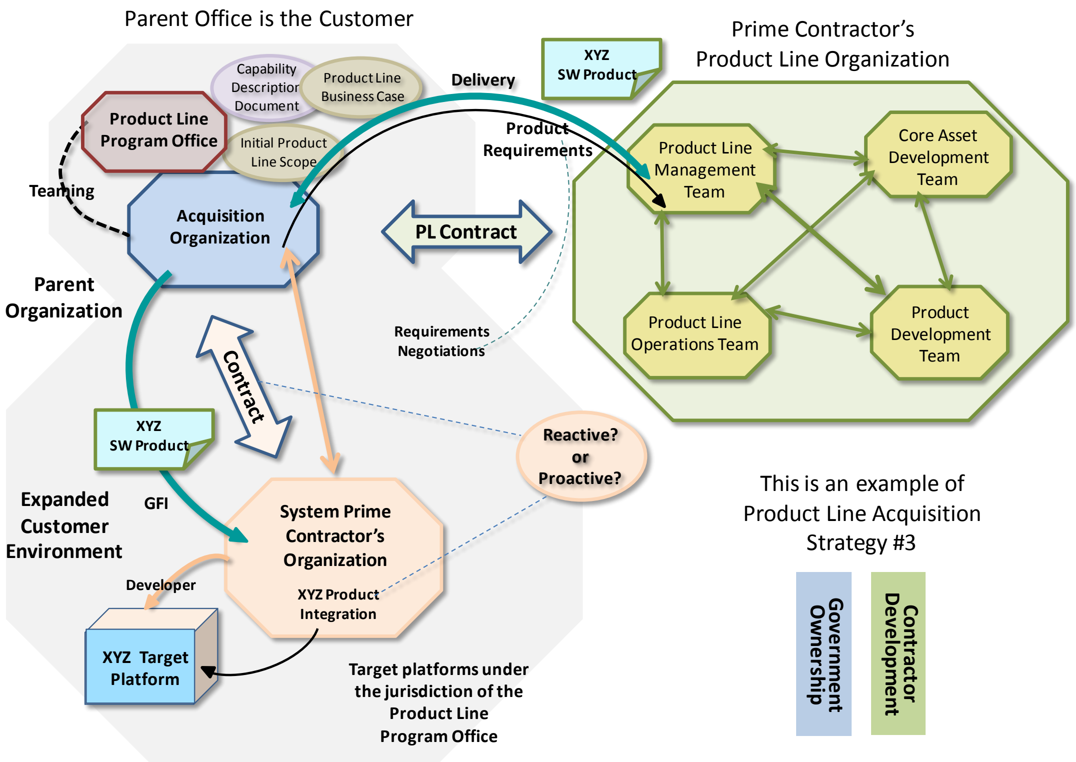

Figure 4: Simplest Case of Customer Interaction with Product Line Developer

In summary, an enterprise view provides a nice basis for describing a product line initiative from an acquisition perspective. This view enables stakeholders to have greater insight and understanding of what a product line acquisition actually entails and is useful for, including

- determining the division of responsibilities between the program office, acquisition organization, and development organization (e.g., contractor)
- understanding stakeholder interactions and interdependencies and assigning specific roles and responsibilities (e.g., requirements management)
- understanding the "contracting realities" of different candidate approaches that are typically glossed over and become problematic downstream unless they are addressed up front

2.3 Joint Fires Product Line – Glen Loupe, PEO STRI

The Joint Fires Product Line (JFPL) is a set of software- and hardware-intensive training systems for cross-service fire control virtual training developed by PEO Simulation Training and Instrumentation (STRI). The product line includes features for training missions that are provided by specific fire control products. To date, these products include several versions of the Joint Terminal Attack Controller (JTAC) trainer and are targeted in the future for the Call for Fire Trainer (CFFT) and the Special Operation Forces Air Ground Simulation (SAGIS). These systems are based on a documented software architecture with reusable software modules and other core assets. The product line core assets also include many guidance, support, and configuration documents, as well as user manuals and catalogs of products, and software core assets. Collectively, these documents capture the production plans for how core assets are used in building products.

All software is government owned and non-proprietary. Software core assets were either mined from legacy systems and brought into the product line, or were created for current product line systems. In most cases, legacy software addressed functionality that had to be brought forward from previous systems into new systems. Some software is built using proprietary commercial off-the shelf (COTS) software but is handled separately from other core assets. Two contractors perform development work on the software architecture.

The product line offers several types of variation. For example, the JTAC trainer supports multiple training formats with the same hardware and software. There are variants for

- multiple customers that cross service boundaries (joint)
- multiple delivery environments
  - portable: easily moved to training locations, to support the needs of joint forces
  - classroom: individual and team training
  - immersive: individual and team training in an environment enforcing tactile training utilizing simulated military equipment
- different numbers of users
- interoperation with other systems to simulate battlefield systems operations

The Joint Fires and Effects Trainer System (JFETS) served as the basis for the product line functional requirements. The trainer supports collective training of fires integration and satisfies training needs at the tactical level for the individual and battle staff. The JFETS legacy functionality is being ported to the JFPL architecture. Development of these follow-on systems has also brought components with new capabilities into the product line.

The product line effort is currently two years old. The first product was delivered in March 2009. Although not entirely faithful to the JFPL concept—getting the product out to the field rapidly was the major impetus—the product provided the needed training capabilities. The risk of not having the funding to correct product line shortcomings in these circumstances is mitigated by having multiple funding sources.

2.3.1 Product Composition and Documentation

A product in the JFPL is composed from complementary hierarchies of software and hardware elements. At the lowest level of the software hierarchy are the services, components, and user interface capabilities. These are composed into *features* that in turn can be composed into *purposes*[3] such as an instructor station, trainee station, single-channel direct view, three-channel direct view, or other. Finally, one or more purposes define a fielded product. Purposes are not core assets—they are used to determine at runtime what the purpose (e.g., an instructor station) of the computer will be. On the hardware side, the lowest level of the compositional hierarchy has *sub-assemblies*. Aggregates of sub-assemblies compose the *assemblies* that in turn are composed into an *environment* for tailoring the physical setup of the product. The hardware hierarchy is mainly built from COTS hardware with standard interfaces, and supports upgrades for technology refresh such as enhanced audio or graphics. Any special or “non-standard” hardware (such as for military equipment) may require specialized interfaces.

[3] Computer software configuration items (CSCIs) exist at the feature or lower level, not at the purpose level.

Variation management is the key to successful reuse and integration of purposes. Product deliveries and variation are measured in part by the number of purposes. To date, these include deliveries of the four purposes mentioned above; two more will be delivered shortly. An XML configuration file defines each purpose as a composition of lower level elements in the hierarchy. These files deal with what is actually running on the product and may be determinable at run-time. For example, a purposes.xml file lists the purposes on a given computer. The configuration lists the features (e.g., control services, equipment tracking, model mapping, close air support missions, or night vision goggles). These features and the components that implement them set up a product for a specific purpose, such as an instructor station. A trainee will also have equipment that varies depending on the service or mission (e.g., night vision goggles, or special operations vs. regular Army operations). This kind of feature variation and composition of component, service, or user interface is an essential aspect of the product line architecture.

How is a JFPL product used? A soldier interacts with training equipment that simulates a battlefield scenario. The scenario may require calling in artillery via close air support, so the training product integrates with equipment such as binoculars, radios, or other systems. In addition to the XML configuration file for a purpose, an XML file is also used to set up the training scenario. Very little “hard coding” is involved.

The JFPL documentation is an integrated collection of electronic content that includes the product line description document, the architecture framework, the architecture specification, a software development kit (SDK), and an SDK guide. There are also catalogs of products, purposes, features, and environments, and related configuration documents for each. Summary information from these documents can be electronically integrated as a product user manual. A goal of the project is to fully automate the generation of the JFPL documentation based on the configuration file approach and also to generate test software.

2.3.2 Concept of Operations

The operation of the JFPL product line organization is focused on managing the core asset base and supporting product development. The government side of the JFPL organization is small, with both core and quality assurance teams supporting system management of the core assets, product line documentation, and catalogs. Activities in system management include providing assets to product builders and managing the acceptance of new products and assets into the product line. Contractor or government development teams are responsible for actual product development. Product development activities range from initial product definition to eventual product development. A “distribution agreement” complemented with a configuration management process governs the transfer of core assets to product developers.

The distribution agreement also governs the integration of products and development artifacts, such as documentation, back into the JFPL. New assets resulting from product development will be incorporated into the JFPL core asset base and made available for future product development efforts. Within this operational concept the role of product line champion is shared, with champions on both the technical and business side.

2.3.3 Adoption Challenges and Lessons Learned

To institutionalize product line practices, the champions of the effort requested that the SEI conduct a Product Line Technical Probe (PLTP)[SM] of the product line organization during July 2009. The probe identified 28 high-priority and 32 lower priority challenges. As of this report, the JFPL organization has completed 16 of the high-priority actions and has all but one of them being worked on. Of the 32 lower priority challenges, 5 have been addressed and actions on 17 of the others are in progress. Some of the major actions under way to address the PLTP findings include

- A software development kit and accompanying guidance documentation are now available.
- A Government configuration management system is being established to manage and maintain software, processes, and documentation.
- The JFPL organization will create a Sharepoint site within the Army Knowledge Online (AKO) site to share information.
- Program management is "marketing" the product line to stakeholders.
- Program management is addressing issues related to product line support funding.

There are still some challenges. For example

- Testing: The JFPL organization has recently established a test policy, but has not fully institutionalized it.
- Funding: The organization is working with program management on securing funding for core asset improvements. The JFPL effort will draw on multiple funding sources (e.g., research and development, Army commands) and will leverage and pool funds.
- Metrics: A full metrics plan is not yet in place. It has been estimated that a training project with a $20 million budget under a single system development approach would cost $3.5 million under the product line approach. Cost savings are being used to implement additional functionality but the JFPL organization does not track concrete numbers yet.

In the JFPL experience, the adoption of a product line approach represents a true paradigm shift in the approach to simulation. Not all contractors have fully endorsed this approach—program management must support a change in the culture and development climate to allow this approach to be adopted.

Finally, Glen Loupe shared some of the lessons learned from the JFPL effort:

- Develop a strong, collective vision.
- Have a champion with appropriate rank and influence.
- Develop a business case or cost-benefit analysis and use this to support funding for maintenance.
- Develop the special testing processes required for shared core assets.

SM Product Line Technical Probe and PLTP are service marks of Carnegie Mellon University.

- Ensure that system program managers take responsibility for the DoD Information Assurance Certification and Accreditation Process (DIACAP). This is not the job of the product line manager.
- Get the SEI involved early.

2.4 Extending the Live Training Transformation Product Line – Bob Becker, Jacobs Engineering

The Army's Live Training Transformation (LT2) product line approach was developed by PEO STRI to apply product line practices and principles in the development of a family of training systems supporting Army live training environments. Different kinds of training systems can be built by reusing the LT2 architecture (which is based on the Common Training Instrumentation Architecture [CTIA]) and the common LT2 components. Common LT2 components include standard interfaces to virtual and constructive simulation systems, tactical command and control systems, and training information systems. Systems in the product line include instrumented live-fire ranges, the Combat Training Center Objective Instrumentation System (CTC-OIS), the Homestation Instrumentation Training System (HITS), and the Military Operations on Urban Terrain Instrumentation System (MOUT-IS).

The live training domain is also a focus of the U.S. Marine Corps (USMC) Range Modernization/Transformation (RM/T) program, which conducted an analysis that showed the program could leverage the capabilities of the Army's LT2 family of training systems. The Army Program Manager – Training Devices (PM TRADE) and the Marine Corps Program Manager – Training Systems (PM TRASYS) signed a program-level agreement to create a partnership with the goals of promoting joint interoperability and reducing acquisition cost and schedule by maximizing reuse of the LT2 product line common components.

The Marine Corps Instrumentation Training System (MC-ITS) is the first target of the new partnership. A mapping of the USMC Range Instrumentation Systems (RIS) Operational Requirements Document (ORD) to the live training domain requirements showed that over 80% of the RIS ORD requirements mapped directly to the live training domain. The commonality of requirements, validated by USMC subject matter experts, laid the foundation for an incremental development strategy for using MC-ITS as the basis for extending the LT2 product line to meet the joint live training needs of PM TRADE and PM TRASYS. Extending the LT2 product line in this way is also part of the longer term consolidated product line management strategy that will reuse LT2 assets in other Marine Corps live training programs.

The MCS-ITS development strategy calls for two incremental releases comprising three software "drops" each. The first increment leverages the Army's Homestation Instrumentation System (HITS) to achieve 87% as-is reuse of common LT2 components. The remaining 13% are a combination of modified LT2 components and new Marine-Corps-specific components. The first software drop for increment 1 occurred in October 2009. Subsequent drops in the first increment will add ground position location information and a distributed interactive simulation (DIS) interface; the second increment will add a high-level architecture (HLA) interface and air position location information.

The major benefits of the product line are reduced costs as well as faster development and deployment. Reusing LT2 components to create MC-ITS has reduced the time to field by an estimated 6 to 12 months, and the reuse strategy is providing two additional software development increments at no additional cost. There are also cost savings associated with integration and test, user training, and sustainability and maintenance.

Other benefits include

- fully government-owned software
- reuse of the LT2 product family infrastructure
- improved exercise planning tools
- interfaces with joint applications

Future work will extend the LT2 product line to include new surrogate training devices for Counter Improvised Explosive Devices (C-IED) and use of the Future Army System of Integrated Targets (FASIT) architecture standard for interactive target integration.

2.5 Common Driver Trainer Product Line – Dean Runzel, PEO STRI

Using simulators to train Army vehicle drivers is an attractive alternative to training on the actual vehicles, which have become increasingly complex and expensive to operate. However, the development and operation of multiple vehicle-specific trainers is not a cost-effective solution—unless there is sufficient commonality across the trainers to enable sharing and reuse of common hardware and software.

In 2004 the Army identified a need to develop a common line of driver training simulators for a range of ground combat vehicles. The goal was to exploit the commonality of training requirements to create common simulator elements, such as the instructor/operator station, motion base, image projectors, and data bases, while factoring out variant items such as the vehicle cab, dashboard, and vehicle dynamics as program-specific elements. The resulting Common Driver Trainer (CDT) product line provides the ability to create 80% of a new driver trainer from the CDT common elements. CDT facilitates the rapid fielding of trainers for a range of vehicles, including the Abrams tank, the Stryker light armored vehicle, and the Mine-Resistant Ambush Protection (MRAP) vehicle.

A typical driver trainer consists of a simulated vehicle cab, instructor/operator station, after action review (AAR) station, visual system, six degrees-of-freedom motion system and a computational system. Interchangeable vehicle cabs and dashboards coupled with vehicle-specific motion cues and realistic terrain databases provide a range of training scenarios. An instructor at the instructor/operator station can monitor a trainee’s performance and also inject emergency situations and vehicle faults into a scenario. Common tasks such as student scoring, review, records management are also handled by the training system.

Fielding CDT as a product line has yielded benefits in the following three areas:

1. Requirements. There is a common system requirements document (SRD) for all trainers. Vehicle-specific variants are covered in appendices.

2. Time to field. A training simulator can be put in the field very quickly—it literally takes more time to “bend the metal” than reuse the software. Use of CDT enabled meeting the aggressive MRAP simulator schedule: 120 days from contract award to first simulator delivery.
3. Component reuse. Eighty percent of a trainer comes from CDT common elements. The biggest factor in vehicle-specific variants is the difference between tracked and wheeled vehicles.

The CDT product line experience has also yielded some issues and lessons learned in the areas of configuration management (CM), personnel, and information assurance certification.

Configuration Management. CM is, according to Dean Runzel, the number-one issue for any product line. Products will usually be at different points in their development cycles and a bug in one product may not necessarily be a problem in another. A solution to a bug in one product won’t necessarily solve the same problem in another product and may even introduce a new problem. The situation is not helped by the fact that most CM tools do not provide adequate support for software product lines. Government and contractor test engineers must understand what constitutes a testable product baseline. Runzel also noted that coordination and communication among team members are key mitigators of CM problems. Because multiple products (variants) usually mean multiple teams, a stable and experienced overarching management team is also vital for the success of the product line.

Personnel. Personnel issues arise because the concept of a product line can be difficult to grasp. The product line approach may require fundamental changes to the tools, techniques, and culture of the multiple disciplines needed for product line success. Personnel changes in government positions have a greater impact on product line programs: There is a relatively small pool of people with product line experience to draw upon when filling vacancies, and newcomers face a significant learning path. Product lines also touch domains beyond just software engineering (other domains include system engineering, information assurance, program management, and contracting). According to Runzel, until product line training is expanded to fields beyond software engineering, product lines will struggle within DoD acquisition programs.

Information Assurance. Current regulations and guidance documents for information assurance don’t even mention product lines. Army Regulation 25-2 for Information Assurance [USAPD 2007], for example, says “All information technology (IT) systems will …” but what constitutes a “system” in the CDT product line context and how it is certified is problematic. There are individual trainers with multiple cabs and dashboards; there is the entire CDT family, which may have multiple configurations, variants, and sub-variants; and there are individual software components. The bottom line, according to Runzel, is that until the people who write regulation and guidance documents become knowledgeable about product lines, compliance will be a struggle.

The CDT product line continues to evolve. A single CDT simulator can have 13 different computers, with half of them running Windows and half running Linux. Some variants support different interfaces for different cabs and are self identifying, so that corresponding drivers and software components are automatically selected and loaded. A remake of the scenario generator is under consideration. Finally, a CDT simulator is no longer tied to a fixed location; there now 13 mobile MRAP variants that can be transported via trailer.

2.6 Common Link Integration Processing Product Line – Gary Newman, Northrop Grumman

The Common Link Integration Processing (CLIP) effort started in 2003 as a joint Navy and Air Force program to produce common software for use on ships and fighters. Currently an Air Force program, CLIP is targeted to support B-1 and B-52 bombers, and F-35 fighters.

CLIP is a software tactical data link (TDL) message processor that can be hosted on multiple weapon system platforms. The purpose of CLIP is to

- reduce cost (ownership, maintenance, and upgrade/refresh)
- improve interoperability by providing common implementations of TDL standards
- eliminate non-program-of-record stovepipe systems

Host platforms have different TDL requirements, interfaces, and operating environments whereby each TDL system deployed defines a unique system that is a member of a family of systems. The CLIP common software is structured by its product line architecture, which was designed to support such qualities as configurability, extensibility and portability. Common software is complemented by a set of developmental artifacts that were used to produce product line systems (instantiations of the product line architecture) customized for specific host platforms such as the B-1B, B-52, F-35, and others. The Broad Area Maritime Surveillance (BAMS) unmanned aircraft system (UAS) is also a near-term candidate for using CLIP.

Gary Newman, the CLIP Chief Architect, gave an overview of the CLIP product line experience from RFP to implementation. He described expectations versus reality, the status of the product line, difficulties and challenges, acquisition aspects, and lessons learned. He explained that a key to meeting program functional and quality requirements is the detailed, open, object-oriented CLIP architecture. The CLIP product line architecture had to be extremely flexible because CLIP was required to interface with multiple host mission computer programs and systems, and run within multiple computing environments.

The motivation for proposing a product line approach was the wording of the original CLIP RFP that required building a "family of systems." However, since the RFP and SOW did not require a product line approach, the contract paid little attention to many product line practice aspects—there was no requirement for contract deliverables such as a product line concept of operations (CONOPS), a product line practice description document, or a product line production plan. These missing items reinforced the lesson that successful product line practice requires the buy-in of both the government and the contractor from the beginning of the acquisition.

Even so, the CLIP program achieved remarkable results because of the underlying product line approach. The requirement to enable use of different data links for different missions means that there are over 41,000 configuration parameters to deal with, mainly because of the many message types. CLIP is able to handle all message formats (and changes to message formats) without affecting or requiring changes to the host platform software. Moreover, any required customization can be done on either the development side or the platform integration side. As a result, Northrop Grumman was able to build a CLIP system for another platform with just 10% of the effort that would previously be required and achieving 94% reuse of existing code.

The CLIP effort is evolving. The current product line engineering processes are under review because although they are documented, they are distributed across several documents; they need to be organized into a more visible and easily accessed central document—a “product line practice description document.” A similar issue exists with the concept of operations; parts of it are distributed within the software engineering management plan, the program management plan, and the software development plan. CLIP management has further identified the need for a production plan to document its current informal product building procedures.

The lessons learned from the CLIP effort to date include the following:

- Product line engineering must be built into the request for proposal (RFP), statement of work (SOW), contract data requirements lists (CDRLs) and the system requirements document (SRD).
- Both the government and the contractors must be ready to pursue product line engineering practices.
- The cost and schedule implications of a product line approach must be considered up front.
- Traceability must be maintained throughout the development artifacts and documents.
- Product line artifacts must be carefully managed by both the government and the contractors.
- A product line approach requires both business and engineering buy-in.

Overall, the CLIP product line is a work in progress. The CLIP team is working on such items as

- software tools to manage variations and configuration parameters
- selective improvements to product line practices
- agreements to feed any product customizations back to core assets
- a more robust product line partnership with the government
- a new paradigm for the sponsor to proactively address funding and schedule impact

These efforts are aimed at keeping the product line vision very much alive so that both Northrop Grumman and the government can reap the full benefits of a product line approach.

3 A Summary of the Facilitated Discussions

Following the presentations, the group participated in a facilitated discussion. Attendees voted on a list of discussion topics; the following were the top three choices.

1. What measures do we need to demonstrate product line value (e.g., cost savings and return on investment)?
2. What's in it for the supplier? What is needed to incentivize suppliers?
   - What incentives can the government provide to encourage suppliers to propose a product line approach even if it is not a hard requirement?
   - What incentives can the government offer to encourage suppliers to respond appropriately to a product line acquisition?
   - What are the barriers from a supplier perspective? How can these barriers be addressed?
3. What technical evaluation criteria (factors and sub-factors) should the government use in evaluating technical proposals and source selection in a product line acquisition?

A summary of each of these discussions follows.

3.1 Discussion Topic: What Measures Do We Need to Demonstrate Product Line Value?

Demonstrating the value of a product lines is frequently thought of as realizing cost savings or a return on investment (ROI), but value may also be tied to product quality or organizational agility. Cost-based measures proposed by the attendees included the following:

- what it cost to produce the product line
- the total life-cycle cost
- cost savings of the product line approach versus the old "stove-piped" approach (measured, for example, in terms of level of effort or person hours to build products using core assets)

There was general agreement on the need to address total cost of ownership over the product line life cycle. An attendee noted that there's already an ROI model, with accompanying tool support, in the Army's Lean Six Sigma effort. Lean Six Sigma can be used as the basis for estimating life-cycle costs of acquisition programs, and you get "double credit" for using it. The caveat is that it is system-focused and would have to be adapted to the needs of a software product line.

Building a realistic business case for a proposed product line also featured prominently in the discussion. The "clone and own" approach is attractive from a development cost perspective, so the business case really needs to look at the life-cycle costs. Cost savings can be estimated by comparing the product line approach against the old stove-piped way of building products. The cost of building core assets will need to be amortized across product development. Establishing a baseline cost of the core asset development effort and tracking the reuse of core assets in products is a way of validating the product line business assumptions.

Costs can be very difficult to tease out, though. One attendee proposed a separate contract line item number (CLIN) that explicitly calls out software architecture and other product line artifacts. Another attendee pointed out the effectiveness of supporting the business case by including core asset costs in the work breakdown structure.

Cost isn't the only determinant of value, however. There are other considerations, such as performance, flexibility, and agility in the market. An organization may push to get a product out quicker even if it only represents a 70% solution. Savings can be measured in terms of test and integration time or time to field.

Some non-cost measures proposed were

- reuse measures (e.g., the percentage of the product provided by the core assets versus the percentage of product-specific code)
- quality measures (e.g., defect rate)
- time-based measures (e.g., schedule savings, or agility—the ability to turn out new products quickly in response to changing needs)

Ultimately, measures of value need to take into account the long-term accrual of product line benefits. There is a tension between fielding a product quickly and investing in the overall product line infrastructure. The typical two-year term for a DoD program manager was cited as one of the forces working against the strategic view required for successful product line adoption.

3.2 Discussion Topic: What's in It for the Supplier? What Is Needed to Incentivize Suppliers?

As in the previous workshop, protection of intellectual property (IP) rights came up in the discussion of incentives. Obtaining the right to license core assets, for example, and reap the benefits from their use and reuse was regarded as an important incentive for suppliers. The general feeling was that a contractor should be free to use the product line approach and the core assets to bid on product development efforts for other organizations.

On the government side, selectively making partial awards in areas of demonstrated expertise and innovation was cited as an incentive. Another approach is to provide incentives around key performance factors—performance-based task orders on indefinite delivery/indefinite quantity (IDIQ) contracts, for example. Overall, the ground rules need to be clearly spelled out up front, so that every supplier is treated identically. Programs also need to codify their commitment to follow a product line approach and not renege on that commitment.

The biggest barrier for suppliers is that the government typically has not rewarded contractors that follow a product line approach. Suppliers are often forced to cover costs specific to software product line needs; some contracts do not permit software product line activities to affect costs. The result is that suppliers must often use their own internal research and development funds to bootstrap the product line approach. As noted at last year's workshop, BAE Systems was successful with exactly this approach. The incentive was the ability to win contracts based on improved performance for the government. Internal research and development (IRAD) funds earned from new contracts permitted the funding of continuous improvement of product line capabilities.

One mitigation strategy for this supplier barrier is to have the government fund the initial core asset development (as in the case of the JFPL) and then bring in the industry base once the core assets have been established. Again, though, the ground rules need to be spelled out, because some contractors would rather stick with a known approach and known problems than risk taking on code with unknown defects.

Another mitigation strategy is to have the government product line organization pay a license fee for every use of a component that a supplier developed from its own funds at its own risk. The developer would retain the IP rights to the component and the government would provide the component as government off-the-shelf (GOTS) software.

One attendee mentioned a Navy submarine program that used small business contractors to innovative software development to demonstrate new capabilities, and then had a prime product line contractor turn selected innovations into products. This led to a discussion of how ownership of the core assets affects the value proposition for product lines. Depending on whether the government, the supplier, or a government-supplier consortium owns the core assets, there will be a different outcome for the associated cost-benefit analysis.

3.3 Discussion Topic: What Technical Evaluation Criteria Should the Government Use?

Ideas for evaluating technical proposals and making source selection decisions in a product line acquisition clustered into categories that can best be characterized in terms of three basic questions.

1. Does a solution exist already?
2. How does the supplier intend to proceed?
3. What is the evidence of supplier competence?

Does a solution exist already? In the spirit of not re-inventing the wheel, it should be possible to examine a proposal in light of existing product lines that already meet (or nearly so) the criteria of the acquisition under consideration. To this end, it would be useful to have a database of known product lines in the DoD. The suggestion was that the government (or the SEI) maintain such a "canonical list" and have it function as a kind of product line registry to be consulted during proposal evaluations.

How does the supplier intend to proceed? The supplier should be required to spell out the technical specifics of how configuration management and variation management will be handled across the product line. In the words of one attendee, "if they can't explain how they do it, they don't really understand product lines." Another suggestion was to ask, in the RFP, for a description of how the supplier built an existing product line. The supplier's response to the RFP would be required to explain, for example, the product line business justification, adoption plan, concept of operations, architecture, and management of variation. The goal is for the government to be able to build evaluation criteria based on prerequisite product line practices. A supplier that has already implemented these practices successfully is in a better position to produce evidence of product line competence.

What is the evidence of supplier competence? The suggestion here was to ask if the supplier had undergone an SEI Product Line Technical Probe (PLTP)[SM] in the previous three years, and, if so, what the supplier had done about the results. Another suggestion, which also relates to how the supplier intends to proceed, was to ask if the supplier had ever built a successful product line and could produce data quantifying the results.

The discussion of supplier evaluation criteria also yielded some caveats. Chief among them was the risk of burdening a supplier with too many requirements or constraints, especially if that supplier is not being paid for what is being asked. A risk mitigation strategy in this case would be to choose from among several contractors based on specific development work done during a down select (which would require some investment dollars from the program office). There was also a suggestion to conduct a PLTP and use the results as intermediate down-select criteria.

Two final observations transcend both government contracting and product lines. The first observation is that past performance of a supplier on a product line effort may not be sufficient; large organizations may be able to do a bait and switch simply to win the contract. The second observation is that it is incumbent on both the government and the contractor to *promote* the product line and the benefits it brings to each party.

SM Product Line Technical Probe and PLTP are service mark s of Carnegie Mellon University.

4 Summary

The 2010 Army Software Product Line Workshop continued the exploration of the challenges and rewards of applying software product line practices within Army programs. The workshop demonstrated a continuance of the trend revealed during the most recent workshops: namely, software product line practice is becoming a reality in the Army and DoD.

Organizations are achieving significant strategic reuse and the associated benefits of reduced cost, reduced schedule, improved quality, and interoperability. There is greater recognition of the importance of cost-benefit analysis for the product line and the necessity of a funding model that provides life-cycle support to the product line. The centrality of an architecture-centric approach to product lines was reaffirmed, as was the use of SEI methods and other architecture methods. Participants also emphasized the importance of organizational cohesion with robust feedback loops between core asset and product development organizations.

Several issues continue to inhibit product line efforts, including

- the difficulty of product line configuration management—keeping core assets and multiple products in parallel development under configuration management
- the need for coordination, communication, and education across *all* stakeholders (the product line enterprise perspective)
- the need for more support for product line approaches from higher levels within the Army/DoD (e.g., acquisition guidance documents and information assurance requirements)

The consensus of the attendees was that the workshop was definitely worthwhile and they would like these workshops to continue. Participants expressed their appreciation to ASSIP, the SEI, and the presenters for their different perspectives and the many “take-aways” from the day’s presentations and discussions. There is particular interest in a workshop specifically on configuration management in software product lines.

Overall, the significant benefits of software product lines in the DoD have been proven, but there is still a need for upper-level DoD support to make product line approaches less of an exception. To that end there is a petition to communicate the product line message to higher levels within the DoD. (The intent is to show how the product line approach is contributing to success, not to force a DoD product line mandate.)

Finally, getting the product line message out means influencing everyone involved in the acquisition process (contracting people, lawyers, program managers, etc.). This is an area where ASSIP can really help, because the strategic focus of ASSIP complements the strategic nature of product line adoption.

If you have any comments on this report or are using a product line approach in the development or acquisition of software-intensive systems for the DoD and would like to participate in a future workshop, please send email to Linda Northrop at lmn@sei.cmu.edu.

References/Bibliography

URLs are valid as of the publication date of this document.

[Bass 1997]
Bass, L.; Clements, P.; Cohen, S.; Northrop, L.; & Withey, J. *Product Line Practice Workshop Report* (CMU/SEI-97-TR-003, ADA327610). Software Engineering Institute, Carnegie Mellon University, 1997.
http://www.sei.cmu.edu/reports/97tr003.pdf

[Bass 1998]
Bass, L.; Chastek, G.; Clements, P.; Northrop, L.; Smith, D.; & Withey, J. *Second Product Line Practice Workshop Report* (CMU/SEI-98-TR-015, ADA354691). Software Engineering Institute, Carnegie Mellon University, 1998.
http://www.sei.cmu.edu/reports/98tr015.pdf

[Bass 1999]
Bass, L.; Campbell, G.; Clements, P.; Northrop, L.; & Smith, D. *Third Product Line Practice Workshop Report* (CMU/SEI-99-TR-003, ADA361391). Software Engineering Institute, Carnegie Mellon University, 1999.
http://www.sei.cmu.edu/reports/99tr003.pdf

[Bass 2000]
Bass, L.; Clements, P.; Donohoe, P.; McGregor, J.; & Northrop, L. *Fourth Product Line Practice Workshop Report* (CMU/SEI-2000-TR-002, ADA375843). Software Engineering Institute, Carnegie Mellon University, 2000.
http://www.sei.cmu.edu/reports/00tr002.pdf

[Bergey 1998]
Bergey, J.; Clements, P.; Cohen, S.; Donohoe, P.; Jones, L.; Krut, R.; Northrop, L.; Tilley, S.; Smith, D.; & Withey, J. *DoD Product Line Practice Workshop Report* (CMU/SEI-98-TR-007, ADA346252). Software Engineering Institute, Carnegie Mellon University, 1998.
http://www.sei.cmu.edu/reports/98tr007.pdf

[Bergey 1999]
Bergey, J.; Campbell, G.; Clements, P.; Cohen, S.; Jones, L.; Krut, R.; Northrop, L.; & Smith, D. *Second DoD Product Line Practice Workshop Report* (CMU/SEI-99-TR-015, ADA375845). Software Engineering Institute, Carnegie Mellon University, 1999.
http://www.sei.cmu.edu/reports/99tr015.pdf

[Bergey 2000a]
Bergey, J.; Campbell, G.; Cohen, S.; Fisher, M.; Jones, L.; Krut, R.; Northrop, L.; O’Brien, W.; Smith, D.; & Soule, A. *Third DoD Product Line Practice Workshop Report* (CMU/SEI-2000-TR-024, ADA387259). Software Engineering Institute, Carnegie Mellon University, 2000.
http://www.sei.cmu.edu/reports/00tr024.pdf

[Bergey 2000b]
Bergey, J.; Fisher, M.; Gallagher, B.; Jones, L.; & Northrop, L. *Basic Concepts of Product Line Practice for the DoD* (CMU/SEI-2000-TN-001, ADA375859). Software Engineering Institute, Carnegie Mellon University, 2000.
http://www.sei.cmu.edu/reports/00tn001.pdf

[Bergey 2001]
Bergey, J. et al. *Fourth DoD Product Line Practice Workshop Report* (CMU/SEI-2001-TR-017, ADA399205). Software Engineering Institute, Carnegie Mellon University, 2001.
http://www.sei.cmu.edu/reports/01tr017.pdf

[Bergey 2003]
Bergey, J.; Cohen, S.; Fisher, M.; Jones, L.; Northrop, L.; & O'Brien, W. *Fifth DoD Product Line Practice Workshop Report* (CMU/SEI-2003-TR-007, ADA416733). Software Engineering Institute, Carnegie Mellon University, 2003.
http://www.sei.cmu.edu/reports/03tr007.pdf

[Bergey 2004]
Bergey, J.; Cohen, S.; Jones, L.; & Smith, D. *Software Product Lines: Experiences from the Sixth DoD Software Product Line Workshop* (CMU/SEI-2004-TN-011). Software Engineering Institute, Carnegie Mellon University, 2004.
http://www.sei.cmu.edu/reports/04tn011.pdf

[Bergey 2005a]
Bergey, J.; Cohen, S.; Donohoe, P.; & Jones, L. *Software Product Lines: Experiences from the Seventh DoD Software Product Line Workshop* (CMU/SEI-2005-TR-001). Software Engineering Institute, Carnegie Mellon University, 2005.
http://www.sei.cmu.edu/reports/05tr001.pdf

[Bergey 2005b]
Bergey, J.; Cohen, S.; Donohoe, P.; & Jones, L. *Software Product Lines: Experiences from the Eighth DoD Software Product Line Workshop* (CMU/SEI-2005-TR-023). Software Engineering Institute, Carnegie Mellon University, Carnegie Mellon University, 2005.
http://www.sei.cmu.edu/reports/05tr023.pdf

[Bergey 2006]
Bergey, John & Cohen, Sholom. *Product Line Acquisition in a DoD Organization–Guidance for Decision Makers* (CMU/SEI-2006-TN-020). Software Engineering Institute, Carnegie Mellon University, 2006.
http://www.sei.cmu.edu/reports/06tn020.pdf

[Bergey 2009]
Bergey, J.; Cohen, S.; Donohoe, P.; Fisher, M.; Jones, L.; & Little, R. *Software Product Lines: Report of the 2009 U.S. Army Software Product Line Workshop* (CMU/SEI-2009-TR-012). Software Engineering Institute, Carnegie Mellon University, 2009.
http://www.sei.cmu.edu/reports/09tr012.pdf

[Chastek 2002]
Chastek, G., ed. *Proceedings of the Second Software Product Line Conference (SPLC2)* (Lecture Notes in Computer Science [LNCS] volume 2379). San Diego, CA, August. 2002. Springer-Verlag, 2002.

[Clements 2001]
Clements, P.; Donohoe, D.; Kyo, K.; McGregor, J.; & Northrop, L. *Fifth Product Line Practice Workshop Report* (CMU/SEI-2001-TR-027, ADA272355). Software Engineering Institute, Carnegie Mellon University, 2001.
http://www.sei.cmu.edu/reports/01tr027.pdf

[Clements 2002]
Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2002.

[Donohoe 2000]
Donohoe, P., ed. *Software Product Lines: Experience and Research Directions, Proceedings of the First Software Product Line Conference (SPLC1)*. Denver, CO, August, 2000. Kluwer Academic Publishers, 2000.

[Geppert 2008]
Geppert, B. & Pohl, K. (eds.) *Proceeding of the 12th International Software Product Line Conference, SPLC 2008*. Limerick, Ireland, September, 2008. Institute of Electrical and Electronics Engineers, Inc., 2008.

[IEEE 2007]
Proceedings of the 11th International Software Product Line Conference, SPLC 2007.
Kyoto, Japan, September, 2007. Institute of Electrical and Electronics Engineers, Inc., 2007.

[McGregor 2009a]
McGregor, John D. “Ecosystem, continued.” *Journal of Object Technology, 8*, 7 (November-December 2009): 7–23.
http://www.jot.fm/issues/issue_2009_11/column1/

[McGregor 2009b]
McGregor, John D. (ed.) *Proceedings of the 13th International Software Product Line Conference, SPLC 2009*. San Francisco, CA, September, 2009. IEEE Computer Society, 2009 (ISBN 978-0-9786956-2-0).

[Nord 2004]
Nord, R., ed. *Proceedings of the Third International Software Product Line Conference (SPLC 2004)* (Lecture Notes in Computer Science [LNCS] volume 3154). Boston, MA, August-September, 2004. Springer-Verlag, 2004.

[Northrop 2008]
Northrop, L. *Software Product Line Essentials*.
http://www.sei.cmu.edu/library/assets/spl-essentials.pdf (2008)

[Northrop 2010]
Northrop, L. & Clements, P. *A Framework for Software Product Line Practice, Version 5.0* http://www.sei.cmu.edu/productlines/framework.html (2010)

[Obbink 2005]
Obbink, H.; & Pohl, K. *Software Product Lines, Ninth International Conference, SPLC 2005 Proceedings*, (Lecture Notes in Computer Science [LNCS] volume 3714). Rennes, France, September, 2005. Springer-Verlag, 2005.

[O'Brien 2006]
O'Brien. L (ed.). *Proceedings of the 10th International Software Product Line Conference, SPLC 2006*. Baltimore, MD, August, 2006. IEEE Computer Society Press, 2006.

[USAPD 2007]
United States Army Publishing Directorate. Army Regulation (AR) 25-2, *Information Assurance*, 2007.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. **AGENCY USE ONLY** (Leave Blank)	2. **REPORT DATE** June 2010	3. **REPORT TYPE AND DATES COVERED** Final

4. **TITLE AND SUBTITLE** Software Product Lines: Report of the 2010 US Army Software Product Line Workshop	5. **FUNDING NUMBERS** FA8721-05-C-0003
6. **AUTHOR(S)** John K. Bergey, Gary Chastek, Sholom Cohen, Patrick Donohoe, Lawrence G. Jones, Linda Northrop	
7. **PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	8. **PERFORMING ORGANIZATION REPORT NUMBER** CMU/SEI-2010-TR-014
9. **SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	10. **SPONSORING/MONITORING AGENCY REPORT NUMBER** ESC-TR-2010-014
11. **SUPPLEMENTARY NOTES**	
12A **DISTRIBUTION/AVAILABILITY STATEMENT** Unclassified/Unlimited, DTIC, NTIS	12B **DISTRIBUTION CODE**

13. **ABSTRACT (MAXIMUM 200 WORDS)**

The Carnegie Mellon Software Engineering Institute held the U.S. Army Software Product Line Workshop on February 11, 2010. The workshop was a hands-on meeting to share Army and Department of Defense product line practices, experiences, and issues and to discuss specific product line practices and operational accomplishments. Participants reported encouraging progress on Army software product lines. This report synthesizes the workshop presentations and discussions.

14. **SUBJECT TERMS** software product line, Army Software Product Line Workshop, DoD product line acquisition	15. **NUMBER OF PAGES** 39
16. **PRICE CODE**	

17. **SECURITY CLASSIFICATION OF REPORT** Unclassified	18. **SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	19. **SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	20. **LIMITATION OF ABSTRACT** UL